Terms used
pipeline and **microinstruction** and **thread** and **decode** and **instruction** and **retire** and **memory** and **counter**

Found **12,306** of **160,906**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 The 16-fold way: a microparallel taxonomy

Barton J. Sano, Alvin M. Despain

December 1993 **Proceedings of the 26th annual international symposium on Microarchitecture**

Full text available: pdf(1.18 MB) Additional Information: full citation, references

Keywords: computer taxonomy, microparallel machines, static and dynamic behavior

2 Difficult-path branch prediction using subordinate microthreads

Robert S. Chappell, Francis Tseng, Adi Yoaz, Yale N. Patt

May 2002 **ACM SIGARCH Computer Architecture News**, Volume 30 Issue 2

Full text available: pdf(1.14 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Branch misprediction penalties continue to increase as microprocessor cores become wider and deeper. Thus, improving branch prediction accuracy remains an important challenge. Simultaneous Subordinate Microthreading (SSMT) provides a means to improve branch prediction accuracy. SSMT machines run multiple, concurrent microthreads in support of the primary thread. We propose to dynamically construct microthreads that can speculatively and accurately pre-compute branch outcomes along frequently mis ...

Keywords: high performance microprocessor, branch prediction, SSMT, SMT, helper thread, microarchitecture, microthread

3 Session 3: Energy-aware OS's: The benefits of event: driven energy accounting in power-sensitive systems

Frank Bellosa

September 2000 **Proceedings of the 9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system**

Full text available: pdf(86.80 KB) Additional Information: full citation, abstract, references, citings

A prerequisite of energy-aware scheduling is precise knowledge of any activity inside the computer system. Embedded hardware monitors (e.g., processor performance counters) have

proved to offer valuable information in the field of performance analysis. The same approach can be applied to investigate the energy usage patterns of individual threads. We use information about active hardware units (e.g., integer/floating-point unit, cache/memory interface) gathered by event counters to establish a t ...

4 A survey of processors with explicit multithreading

Theo Ungerer, Borut Robič, Jurij Šilc

March 2003 **ACM Computing Surveys (CSUR)**, Volume 35 Issue 1

Full text available: pdf(920.16 KB) Additional Information: full citation, abstract, references, citings, index terms

Hardware multithreading is becoming a generally applied technique in the next generation of microprocessors. Several multithreaded processors are announced by industry or already into production in the areas of high-performance microprocessors, media, and network processors.A multithreaded processor is able to pursue two or more threads of control in parallel within the processor pipeline. The contexts of two or more threads of control are often stored in separate on-chip register sets. Unused i ...

Keywords: Blocked multithreading, interleaved multithreading, simultaneous multithreading

5 Transient fault detection via simultaneous multithreading

Steven K. Reinhardt, Shubhendu S. Mukherjee

May 2000 **ACM SIGARCH Computer Architecture News , Proceedings of the 27th annual international symposium on Computer architecture**, Volume 28 Issue 2

Full text available: pdf(151.56 KB) Additional Information: full citation, abstract, references, citings, index terms

Smaller feature sizes, reduced voltage levels, higher transistor counts, and reduced noise margins make future generations of microprocessors increasingly prone to transient hardware faults. Most commercial fault-tolerant computers use fully replicated hardware components to detect microprocessor faults. The components are lockstepped (cycle-by-cycle synchronized) to ensure that, in each cycle, they perform the same operation on the same inputs, producing the same outputs in the abs ...

6 Relational profiling: enabling thread-level parallelism in virtual machines

Timothy Heil, James E. Smith

December 2000 **Proceedings of the 33rd annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(237.19 KB) ps(1.61 MB) Publisher Site Additional Information: full citation, references, citings, index terms

7 A dynamic multithreading processor

Haitham Akkary, Michael A. Driscoll

November 1998 **Proceedings of the 31st annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(2.67 MB) Additional Information: full citation, references, citings, index terms

8 ProfileMe: hardware support for instruction-level profiling on out-of-order processors

Jeffrey Dean, James E. Hicks, Carl A. Waldspurger, William E. Weihl, George Chrysos

December 1997 **Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture**

Full text available:

pdf(1.60 MB) Additional Information: full citation, abstract, references, citings, index terms
Publisher Site

Profile data is valuable for identifying performance bottlenecks and guiding optimizations. Periodic sampling of a processor's performance monitoring hardware is an effective, unobtrusive way to obtain detailed profiles. Unfortunately, existing hardware simply counts events, such as cache misses and branch mispredictions, and cannot accurately attribute these events to instructions, especially on out-of-order machines. We propose an alternative approach, called ProfileMe, that samples instructio ...

9 Tolerating late memory traps in ILP processors

Xiaogang Qiu, Michel Dubois

May 1999 **ACM SIGARCH Computer Architecture News , Proceedings of the 26th annual international symposium on Computer architecture**, Volume 27 Issue 2

Full text available: pdf(100.18 KB) Additional Information: full citation, abstract, references, citings, index terms
Publisher Site

ILP processors can execute a large number of instructions at the same time. Thus it becomes more and more difficult to support traps efficiently. On the other hand a current trend in architecture is to support various memory functions in software rather than hardware, usually by trapping the execution processor on a cache miss, TLB miss or a failed access to a local or remote memory. These late memory traps block the faulting instruction at the top of the active list, backing up the pipeline. Mo ...

10 Continual flow pipelines

Srikanth T. Srinivasan, Ravi Rajwar, Haitham Akkary, Amit Gandhi, Mike Upton

October 2004 **Proceedings of the 11th international conference on Architectural support for programming languages and operating systems**, Volume 38 , 39 , 32 Issue 5 , 11 , 5

Full text available: pdf(274.26 KB) Additional Information: full citation, abstract, references, index terms

Increased integration in the form of multiple processor cores on a single die, relatively constant die sizes, shrinking power envelopes, and emerging applications create a new challenge for processor architects. How to build a processor that provides high single-thread performance and enables multiple of these to be placed on the same die for high throughput while dynamically adapting for future applications? Conventional approaches for high single-thread performance rely on large and complex co ...

Keywords: CFP, instruction window, latency tolerance, non-blocking

11 Special session on memory wall: Fighting the memory wall with assisted execution

Michel Dubois

April 2004 **Proceedings of the 1st conference on Computing frontiers**

Full text available: pdf(231.18 KB) Additional Information: full citation, abstract, references, index terms

Assisted execution is a form of simultaneous multithreading in which a set of auxiliary "assistant" threads, called *nanothreads*, is attached to each thread of an application. Nanothreads are lightweight threads which run on the same processor as the main (application) thread and help execute the main thread as fast as possible. Nanothreads exploit resources that are idled in the processor because of hazards due to program dependencies and memory access delays.Assisted execution has the po ...

Keywords: cache memories, latency tolerance, prefetching, simultaneous multithreading, superscalar processors

12 SMTp: An Architecture for Next-generation Scalable Multi-threading
Mainak Chaudhuri, Mark Heinrich
March 2004 **ACM SIGARCH Computer Architecture News , Proceedings of the 31st annual international symposium on Computer architecture ISCA '04**, Volume 32 Issue 2
Full text available: pdf(247.58 KB) Additional Information: full citation, abstract

We introduce the SMTp architecture-an SMT processoraugmented with a coherence protocol thread context,that together with a standard integrated memory controllercan enable the design of (among other possibilities) scalablecache-coherent hardware distributed shared memory(DSM) machines from commodity nodes. We describe theminor changes needed to a conventional out-of-order multi-threadedcore to realize SMTp, discussing issues related toboth deadlock avoidance and performance. We then compareSMTp p ...

13 The instruction parsing microarchitecture of the CVAX microprocessor
David W. Archer
December 1987 **Proceedings of the 20th annual workshop on Microprogramming**
Full text available: pdf(682.89 KB) Additional Information: full citation, abstract, references

CVAX is a single chip, CMOS VLSI VAX microprocessor. Several microarchitectural innovations helped achieve the desired performance goal of this machine. In particular, the instruction parsing and prefetching mechanism is different from other VAX implementations. This new instruction parsing microarchitecture is discussed in this paper.

14 Converting thread-level parallelism to instruction-level parallelism via simultaneous multithreading
Jack L. Lo, Joel S. Emer, Henry M. Levy, Rebecca L. Stamm, Dean M. Tullsen, S. J. Eggers
August 1997 **ACM Transactions on Computer Systems (TOCS)**, Volume 15 Issue 3
Full text available: pdf(526.39 KB) Additional Information: full citation, abstract, references, citings, index terms, review

To achieve high performance, contemporary computer systems rely on two forms of parallelism: instruction-level parallelism (ILP) and thread-level parallelism (TLP). Wide-issue super-scalar processors exploit ILP by executing multiple instructions from a single program in a single cycle. Multiprocessors (MP) exploit TLP by executing different threads in parallel on different processors. Unfortunately, both parallel processing styles statically partition processor resources, thus preventing t ...

Keywords: cache interference, instruction-level parallelism, multiprocessors, multithreading, simultaneous multithreading, thread-level parallelism

15 Slipstream processors: improving both performance and fault tolerance
Karthik Sundaramoorthy, Zach Purser, Eric Rotenburg
November 2000 **Proceedings of the ninth international conference on Architectural support for programming languages and operating systems**, Volume 28 , 34 Issue 5 , 5
Full text available: pdf(111.54 KB) Additional Information: full citation, abstract, references, citings, index terms

Processors execute the full dynamic instruction stream to arrive at the final output of a program, yet there exist shorter instruction streams that produce the same overall effect. We propose creating a shorter but otherwise equivalent version of the original program by removing ineffectual computation and computation related to highly-predictable control flow. The shortened program is run concurrently with the full program on a chip multiprocessor simultaneous multithreaded processor, with two ...

16 Slipstream processors: improving both performance and fault tolerance
Karthik Sundaramoorthy, Zach Purser, Eric Rotenberg

November 2000 **ACM SIGPLAN Notices**, Volume 35 Issue 11

Full text available: pdf(1.51 MB) Additional Information: full citation, abstract, references, index terms

Processors execute the full dynamic instruction stream to arrive at the final output of a program, yet there exist shorter instruction streams that produce the same overall effect. We propose creating a shorter but otherwise equivalent version of the original program by removing ineffectual computation and computation related to highly-predictable control flow. The shortened program is run concurrently with the full program on a chip multiprocessor or simultaneous multithreaded processor, with t ...

17 Instruction fetch mechanisms for multipath execution processors

Artur Klauser, Dirk Grunwald

November 1999 **Proceedings of the 32nd annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(1.43 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Branch mispredictions can have a major performance impact on high-performance processors. Multipath execution has recently been introduced to help limit the misprediction penalties incurred by branches that are difficult to predict. This paper presents efficient instruction fetch architecture designs for these multipath processor execution cores. We evaluate a number of design trade-offs for the first-level instruction cache and the multipath PC fetch arbiter. Furthermore we evaluate the e ...

18 Superscalar architectures: Dual use of superscalar datapath for transient-fault detection and recovery

Joydeep Ray, James C. Hoe, Babak Falsafi

December 2001 **Proceedings of the 34th annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(1.18 MB) Publisher Site Additional Information: full citation, abstract, references, citings

Diminutive devices and high clock frequency of future microprocessor generations are causing increased concerns for transient soft failures in hardware, necessitating fault detection and recovery mechanisms even in commodity processors. In this paper, we propose a fault-tolerant extension for modern superscalar out-of-order datapath that can be supported by only modest additional hardware. In the proposed extensions, error-detection is achieved by verifying the redundant results of dynamically r ...

19 Toward kilo-instruction processors

Adrián Cristal, Oliverio J. Santana, Mateo Valero, José F. Martínez

December 2004 **ACM Transactions on Architecture and Code Optimization (TACO)**, Volume 1 Issue 4

Full text available: pdf(1.16 MB) Additional Information: full citation, abstract, references, index terms

The continuously increasing gap between processor and memory speeds is a serious limitation to the performance achievable by future microprocessors. Currently, processors tolerate long-latency memory operations largely by maintaining a high number of in-flight instructions. In the future, this may require supporting many hundreds, or even thousands, of in-flight instructions. Unfortunately, the traditional approach of scaling up critical processor structures to provide such support is impractica ...

Keywords: Memory wall, instruction-level parallelism, kilo-instruction processors, multicheckpointing

20 Method-level phase behavior in java workloads

Andy Georges, Dries Buytaert, Lieven Eeckhout, Koen De Bosschere

October 2004 **ACM SIGPLAN Notices , Proceedings of the 19th annual ACM SIGPLAN Conference on Object-oriented programming, systems, languages, and applications**, Volume 39 Issue 10

Full text available: pdf(695.63 KB) Additional Information: full citation, abstract, references, index terms

Java workloads are becoming more and more prominent on various computing devices. Understanding the behavior of a Java workload which includes the interaction between the application and the virtual machine (VM), is thus of primary importance during performance analysis and optimization. Moreover, as contemporary software projects are increasing in complexity, automatic performance analysis techniques are indispensable. This paper proposes an off-line method-level phase analysis approach for ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

	Type	L #	Hits	Search Text	DBs	Time Stamp
1	IS&R	L1	2	("6,493,741").PN.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 14:14
2	BRS	L2	0	(paus$ or prevent$) near3 insruction	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 14:16
3	BRS	L3	0	(pausing or preventing) near3 insruction	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 14:16
4	BRS	L4	12421	(paus$ or prevent$) near3 thread	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 14:16
5	BRS	L5	264	4 and pipeline	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 14:24
6	BRS	L6	25	5 and microinstruction	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 14:25

7	BRS	L7	24	6 and retire$	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 15:58

	Type	L #	Hits	Search Text	DBs	Time Stamp
8	BRS	L8	23	7 and counter	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 15:58
9	BRS	L9	23	7 and (timer or counter)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 15:58
10	BRS	L10	23	9 and decod$	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 15:58
11	BRS	L11	14	10 and resum$	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 15:59
12	IS&R	L12	1302	((712/220,228,244) or (709/107)).CCLS.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 14:24
13	BRS	L13	36	12 and 4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 14:24

14	BRS	L14	26	13 and pipeline	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 16:09

	Type	L #	Hits	Search Text	DBs	Time Stamp
15	BRS	L15	11	14 and microinstruction	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 14:25
16	BRS	L16	11	15 and retire$	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 15:58
17	BRS	L17	11	16 and counter	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 15:58
18	BRS	L18	11	16 and (timer or counter)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 15:58
19	BRS	L19	11	18 and decod$	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 15:58
20	BRS	L20	7	19 and resum$	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 16:08

21	BRS	L21	4	15 not 20	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 16:08

	Type	L #	Hits	Search Text	DBs	Time Stamp
22	BRS	L22	15	14 not 15	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/09/13 16:09